www.ietdl.org

Published in IET Image Processing
Received on 10th June 2009
Revised on 16th March 2011
doi: 10.1049/iet-ipr.2009.0344

ISSN 1751-9659

Very-large-scale integration design of a low-power and cost-effective context-based adaptive variable length coding decoder for H.264/AVC portable applications

H.-J. Huang[1] *C.-H. Fang*[2] *C.-P. Fan*[2]

[1]*Department of Electronics Engineering, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan*
[2]*Department of Electrical Engineering, National Chung Hsing University, 250 Kuo-Kuang Road, Tai-chung 402, Taiwan*
E-mail: cpfan@dragon.nchu.edu.tw

Abstract: Context-based adaptive variable length coding (CAVLC) is a new and efficient entropy coding tool in H.264/AVC (advanced video coding). This study proposes a low-power and cost-effective CAVLC decoding architecture for the H.264/AVC baseline profile. Specifically, this study proposes an optimum two-layer power model for the variable length look-up table (VLUT) in the CAVLC decoder, and divides the decoding phase of the LUT into two-layer decoding to reduce power consumption. To achieve a cost-effective design, the proposed design merges common codewords to reduce the hardware cost among different LUTs in the second layer decoding. The proposed decoder is based on Taiwan Semiconductor Manufacturing Company (TSMC) 0.18 μm CMOS technology, and was completely verified on a field-programmable gate array (FPGA) emulation platform. The proposed design meets the demands of the real-time CAVLC decoding and reduces power consumption by 44–48% more than previous low-power CAVLD schemes. Finally, the proposed low-power and cost-effective CAVLD design is suitable for H.264/AVC portable applications.

1 Introduction

The ISO/IEC MPEG and ITU-T VCEG formed the Joint Video Team in 2001 and established an up-to-date video compression standard in the same year. The H.264/AVC standard [1] was widely adopted by video communication systems in the following years. Although the H.264 standard improved the compression ratio and video quality, it also increased the complexity and power consumption of video compression systems. Variable length coding (VLC) is one of a key technology in video compression systems, and multimedia standards specify many VLC methods. Traditional methods can be assigned to regular probability models. Context-based adaptive variable length coding (CAVLC) is the entropy coding tool of the H.264/AVC baseline profile, and is an adaptive coding tool that follows previous coding data to choose its probability model. Although CAVLC increases the compression ratio, it also increases the power consumption and hardware cost of the decoder. Since battery development has generally lagged behind the demands of mobile equipment (e.g. personal digital assistant (PDA), mobile phones etc.), low-power and cost-effective designs have become major issues. Thus, it is necessary to develop a CAVLC decoder (CAVLD) with a low-power design that also meets the demands of real-time CAVLC decoding for portable applications.

Many real-time video systems, that is HDTV in ATV project, require high-performance decoding. The parallel VLC decoding presented in [2] is suitable for video coding systems. Except for high-performance applications, previous study also proposed some low-power parallel VLD designs for MPEG-2 video coding [3–5]. Cho *et al.* [4] proposed non-uniform fine-gain tables, which use the property of the VLC to reduce the power consumption at the architecture level. Lee and Park [5] then proposed a low-power variable length decoder, stuffing some fixed caches into the variable length code detector to reduce power consumption. These low-power VLD designs [3–5] for MPEG-2 made it possible to develop a low-power CAVLD design for H.264/AVC.

The H.264/AVC decoding system uses CAVLD to decode residual blocks. There are many variable length look-up tables (VLUT) in the CAVLD. These VLUTs can cause many problems, such as higher hardware cost, greater power consumption and low decoding efficiency. Owing to the high decoding complexity of CAVLC decoding, it is more difficult to design an efficient CAVLD than a traditional variable VLD. To solve these problems, previous researchers have proposed various hardware solutions [6–17]. Di *et al.* implemented the fundamental architecture of CAVLD in [6], but this design resulted in a large hardware area. Chang *et al.* [7] proposed a low-cost and high-performance CAVLD architecture. These approaches focused on high-performance CAVLD, but failed to optimise power consumption. Lin [8] and Lin *et al.* [9] proposed a low-power CAVLD architecture that divides the VLUT into some sub-VLUTs based on code length. This design added many latches to the entry of the sub-VLUTs to prevent variance in dynamic power consumption. Although this design reduced CAVLD power consumption, the high number of latches resulted in a relatively high hardware cost.

© The Institution of Engineering and Technology 2012

Kim *et al.* [10] developed a new CAVLC decoding method using arithmetic operations instead of the conventional table look-up method, which required lots of memory accesses. Their algorithm achieves the efficiency required for mobile video environments, in which low-power consumption and low memory usage are essential for multimedia codecs. Since a large number of memory accesses is a serious problem for videophone applications, Moon *et al.* [11] proposed an efficient Coeff_Token (CT) VLD and a new Run_Before (RB) VLD based on arithmetic operations for low-power applications. The arithmetic-based CAVLD architectures in [10, 11] achieved fast decoding and reduced power consumption by decreasing memory accesses. Thus, these designs have the potential for the H.264/AVC portable utilisation.

Wen *et al.* [12] proposed a new high-throughput architecture to realise the context-adaptive variable-length decoder of H.264/AVC baseline profile. They conducted a thorough analysis of the inherent parallelism of the CAVLC algorithm, and adopted a bit-position VLC decoding approach to decode multiple symbols concurrently. Yu and Chang [14] presented a high-performance CAVLC decoding architecture for H.264/AVC. Instead of just skipping the zero block, this design explores the features of the CAVLD decoding process to skip possible processes efficiently if none needs to be decoded. Although these are high throughput CAVLD designs, their power consumption is too high for portable applications.

Sun *et al.* [15] applied the self-grouping algorithm and combined the same basic clusters to reduce the bit numbers needed to decode the symbol data in the VLC-based decoding systems. To achieve the programmability of the VLC decoder, a memory-based architecture with improved memory efficiency was proposed in [16]. This study extends the group-based look-up table algorithm to multi-table merging, which further decreases the redundancy of groups. The multi-table merging algorithm efficiently integrates all coding tables into memory. The look-up table mergence (LTM)-based CAVLD architecture in [15, 16] makes it possible to build a reduced cost CAVLD design for H.264/AVC, and this issue is also important to the portable applications.

Lin *et al.* [18] proposed a high-performance CAVLC decoding architecture for H.264 decoders, and precisely analysed the CAVLC decoding process to improve overall performance. The maximum frequency of this architecture was 213 MHz, which is fast enough to decode the 1080 HD (1920 × 1088) video format. Tsai and Fang [19] proposed an efficient CAVLD algorithm for H.264/AVC baseline profile. They proposed two new methods to improve the throughput of the CAVLCD: multilevel decoding and non-zero skip for RB decoding. By using parallel operations on the level decoder, they increased throughput and achieved sufficient performance for real-time HD1080i decoding. Another study [20] applies the extensive parallel CAVLC decoding architecture with prefix pre-computation to process the decoding operations efficiently. Compared to the previous multiple-symbol parallel decoding, this design reduces the area by 46% while achieving the same performance in throughput. For multi-symbol decoder implementation, Lee *et al.* [21] developed a low-complex very-large-scale integration (VLSI) architecture for CAVLC decoding that increases throughput by breaking the recursive dependency among codewords. This CAVLCD meets the real-time requirements of HD 1920 × 1080 decoding.

Though previous studies [7, 12, 14, 19–21] present high throughput CAVLD designs for real-time HD applications, their hardware cost is still large and the power consumption might not be suitable for H.264 portable applications. Although the efficient CAVLD architectures for H.264/AVC in [8, 9, 18] are low-cost and low-power designs, the goal of this study is to develop a lower power CAVLD design with an effective cost for H.264 portable applications.

To achieve the lower power consumption with a cost-effective hardware for H.264/AVC portable applications, the proposed design uses the optimum two-layer power model of the LUT in the CAVLD and divides the decoding phase of the LUT into two-layer decoding. In the first layer decoding, the short codewords with higher hit probability are used for decoding, and the other codewords with lower hit probability are decoded in parallel in the second layer decoding. Using the same technique in [13, 15, 16], this design merges common codewords to reduce the hardware cost of the different LUTs in the second layer decoding. Thus, the proposed CAVLD architecture achieves further power reduction with a cost-effective hardware.

The rest of this paper is organised as follows. Section 2 discusses the topic of the low-power CAVLD design. Section 3 describes the principles of the proposed low-power and cost-effective architecture for the CAVLD design. Section 4 presents the functional verification, VLSI results and comparisons. Finally, Section 5 presents the conclusion.

2 Motivation

The CAVLD plays the role of entropy decoding in a H.264/AVC decoder. When the variable length bitstream is inputted to the decoder, the CAVLD outputs the residual block information for the transform coding, the spatial prediction information for the spatial compensation, and the motion vector for motion compensation. Then, video frames are reconstructed and displayed as the output of a H.264/AVC decoder. The decoding flow includes CT, Trailing_One, Level, Total_Zeros (TZ) and RB. The hardware blocks of the CAVLD [18] can be divided into four parts: the symbol decoder, the barrel shifter, the control unit and the output buffer. Moreover, the symbol decoder in the CAVLD is composed of CT, Trailing_One, Level, TZ and RB modules. The CT, TZ, RB modules can generally be realised with the table-based architecture, whereas the Trailing_One and Level modules can be implemented with arithmetic operations. In a direct and convenient implementation of the symbol decoder, the table-based part occupies approximately 74.3% of the chip area, whereas the arithmetic part occupies the remaining 25.7%.

Since the functions of CT, TZ and RB occupy most of the chip area in the symbol decoder, reducing the cost and power consumption of these modules is an important issue for portable applications. During the decoding process, the power consumption of these modules can be expressed by the following equation

$$E_{\mathrm{CAVLD,VLUT}} = E_{\mathrm{CT}} + E_{\mathrm{TZ}} + E_{\mathrm{RB}} \tag{1}$$

where $E_{\mathrm{CAVLD,VLUT}}$ is the overall energy of the VLUT in the CAVLD, and E_{CT}, E_{TZ} and E_{RB} are the average energies of the CT, TZ and RB decoding modules, respectively. To achieve the goal of the low-power CAVLD design, it is necessary to derive an optimum power model. In the non-optimal and direct CAVLD design, the area distribution of the overall VLUTs [17] is as follows: the TZ, RB and CT parts occupy 21.8, 7.8 and 70.1% of the area, respectively. Consequently, this study first analyses the power consumption of the CT module, which has a higher hardware cost than the other modules.

© The Institution of Engineering and Technology 2012

Equation (2) in [4] shows the average power consumption for decoding per codeword in one VLUT, where n is the total number of codewords in the VLUT, $P_{\mathrm{CW}i}$ is the searching probability of the codeword i and E_{CW_i} is the decoding energy of the codeword i

$$E_{\mathrm{VLUT,avg}} = \sum_{i=1}^{n} P_{\mathrm{CW}_i} E_{\mathrm{CW}_i} \tag{2}$$

This equation shows that the power consumption of a single VLUT is relatively large because long codewords are always activated during decoding. For this reason, some researchers have focused on partitioning a VLUT into some small sub-VLUTs to reduce power consumption. In this case, the energy model for the architecture in [4] with the hierarchy scheme is as follows

$$\begin{aligned} E_{\mathrm{VLUT,avg}} = {} & P_{r_1} E_{H_1} + P_{r_2}(E_{H_2} + E_{M_1}) \\ & + P_{r_3}(E_{H_3} + E_{M_1} + E_{M_2}) + \cdots \\ & + P_{r_w}\left(E_{H_w} + \sum_{i=1}^{w-1} E_{M_i}\right) + E_{\mathrm{other}}(w) \end{aligned} \tag{3}$$

where w is the total number of sub-VLUTs, P_{r_i} is the probability when the ith sub-VLUT is hit, E_{H_i} is the energy consumption of the ith sub-VLUT when there is a hit and E_{M_i} is the energy consumption for a searching miss at the ith sub-VLUT. The term $E_{\mathrm{other}}(w)$ represents extra energy consumption, which is introduced by the table partition; for instance, the additional MUX, the branch and the latch. Note that the accumulation of miss energy increases as the depth of the hitting sub-VLUT increases.

From the descriptions above, two major factors influence the average energy: E_{H_i} and E_{M_i}. Short codewords are always with the high decoding probability, causing the high probability tables of the first layer to consume the most energy. Therefore E_{H_1} must be low enough to achieve the minimum average energy. Besides, the QP value is a key factor influencing video quality. A low QP value makes the high video quality. In other words, a low QP value may activate the low probability tables more frequently. For this reason, it is necessary to avoid the accumulation of the searching miss energy, E_{M_i}. Section 3.1 states the optimal power model with the proposed two-layer decoding scheme.

3 Proposed low-power and cost-effective VLSI architecture

To achieve a further low-power design, Section 3.1 applies the optimum two-layer low-power model of the LUT and divides the decoding phase of the LUT into two-layer decoding. In the first layer decoding, the short codewords with higher hit probability are decoded. Next, the other long codewords with lower hit probability are decoded in parallel in the second layer decoding to avoid the accumulation of miss energy. To obtain a cost-effective design, Section 3.2 applies the technique of LTM to merge common codewords and reduce the hardware cost.

3.1 Two-layer searching (TLS) architecture

This study proposes a novel architecture of the LUT-based decoder to reduce both the power consumption and the hardware cost. Based on the observations above, the proposed architecture is partitioned into two layers to reduce the accumulation of E_M and the overload area. Some short codewords are decoded in the first layer, whereas longer codewords are decoded in the second layer. When a decoded input symbol is not found in the first layer decoding, the decoding process in the second layer will be enabled to work in the next cycle. This architecture differs from traditional VLD architectures [3, 5, 8, 9] which trace the relationship between the number of the partition and the power consumption to minimise power consumption. Although some architectures [3, 5, 8, 9] can efficiently achieve the power reduction using prefix pre-decoding and table partitioning technologies, they cannot reduce the decoding power further, and then their hardware cost cannot achieve a balance with the power consumption.

To strike the balance between the power and the area, the proposed design adopts a parallel decoding architecture in the second layer decoding. Since the implementation of the parallel decoding is not traced by statistics, there is one critical issue in our architecture. This issue is how to choose and partition the codeword efficiently between the two-layer decoding, that is the high-hit and the low-hit decoding layers.

3.1.1 Table partition method: A general codeword in H.264/AVC can be divided into two parts, that is the variable-length prefix and the 3-bit suffix. In the codeword structure, the prefix code is cascaded by the suffix code. Most long codewords have this structure, and the part of the prefix in long codewords can be merged among the different LUTs to reduce the area for realisations. Thus, the sizes of the searching codeword in the LUT can be reduced to 2 bits or 3 bits. However, the structure of short codewords with high decoding probability may be irregular. The difference between long and short codewords is thus a criterion for two-layer decoding. In other words, irregular short codewords are distributed to the first layer decoding, and the other regular codewords are located in the second layer decoding. The LUTs in the first and the second layers are implemented using the hardwired logic, and not by memory modules. Fig. 1 illustrates the architecture of the two-layer LUTs.

The searching and decoding process is described as follows. When a codeword is not found in the first layer decoding, the prefix decoding will start to decode the prefix of the codeword during the same cycle; otherwise, the prefix decoding will be disabled from decoding to avoid dynamic power consumption. After the prefix decoding, the result of the prefix is stored in the register, and it will be used to choose the correct sub-LUT in the next cycle.

3.1.2 Power consumption analysis: According to the non-optimal power model in (3), there are two important factors that directly influence the power consumption of the LUT architecture: E_{H_1} and E_{M_i}. Since the codewords in the first layer are short, most of the decoding processes will be located here. This study simulates four different types CIF 30 frames/s sequences using the H.264 reference software JM10.2 [22] under the baseline mode. Five different QP values, that is 16, 22, 28, 34 and 40, are chosen for simulations and average analyses. Figs. 2–4 show the results of decoding probabilities in the first and second layers with all intra, IPPPP and IBPBP frames, respectively. Although not all of the searching codewords are decoded in the first layer decoding, the average decoding probability in the first layer is approximately 80% when the QP value exceeds 28. Table 1 lists the number of codewords of the module CT in the first layer decoding. Lin's design moves

© The Institution of Engineering and Technology 2012

doi: 10.1049/iet-ipr.2009.0344

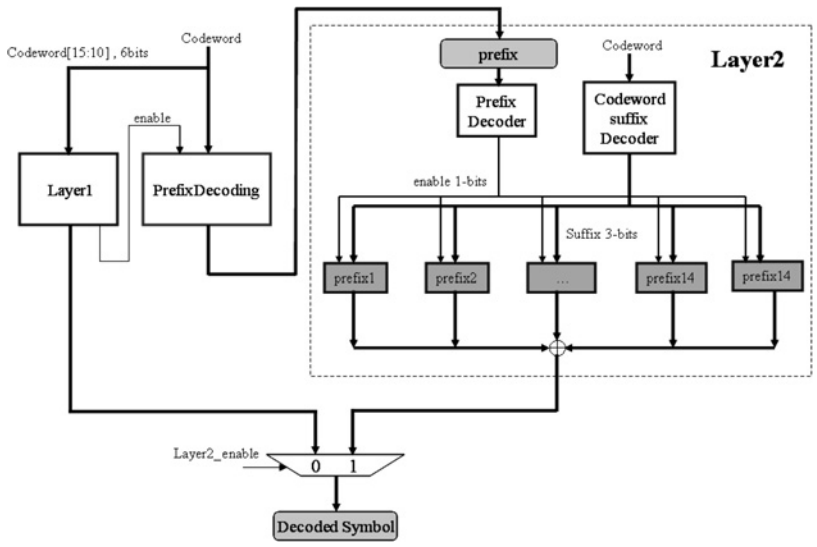

Fig. 1 *Architecture of the proposed two-layer LUTs*

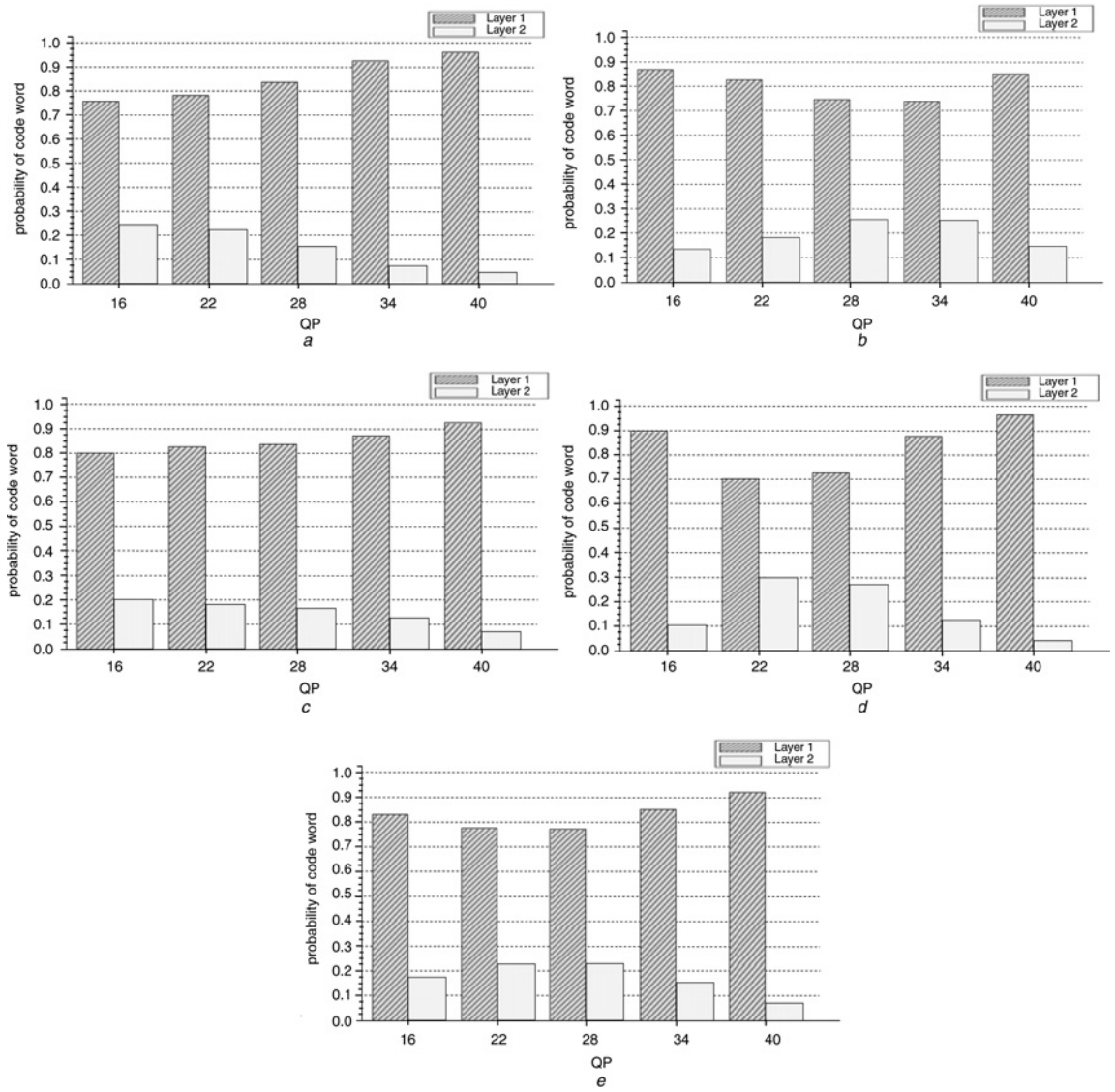

Fig. 2 *Probabilities of layer with the proposed two-layer decoding with all intra frames*

a CIF 30 frames/s 'Foreman' sequence
b CIF 30 frames/s 'Mobile' sequence
c CIF 30 frames/s 'Akiyo' sequence
d CIF 30 frames/s 'Coastguard' sequence
e Average probability of the four CIF 30 frames/s sequences

© The Institution of Engineering and Technology 2012

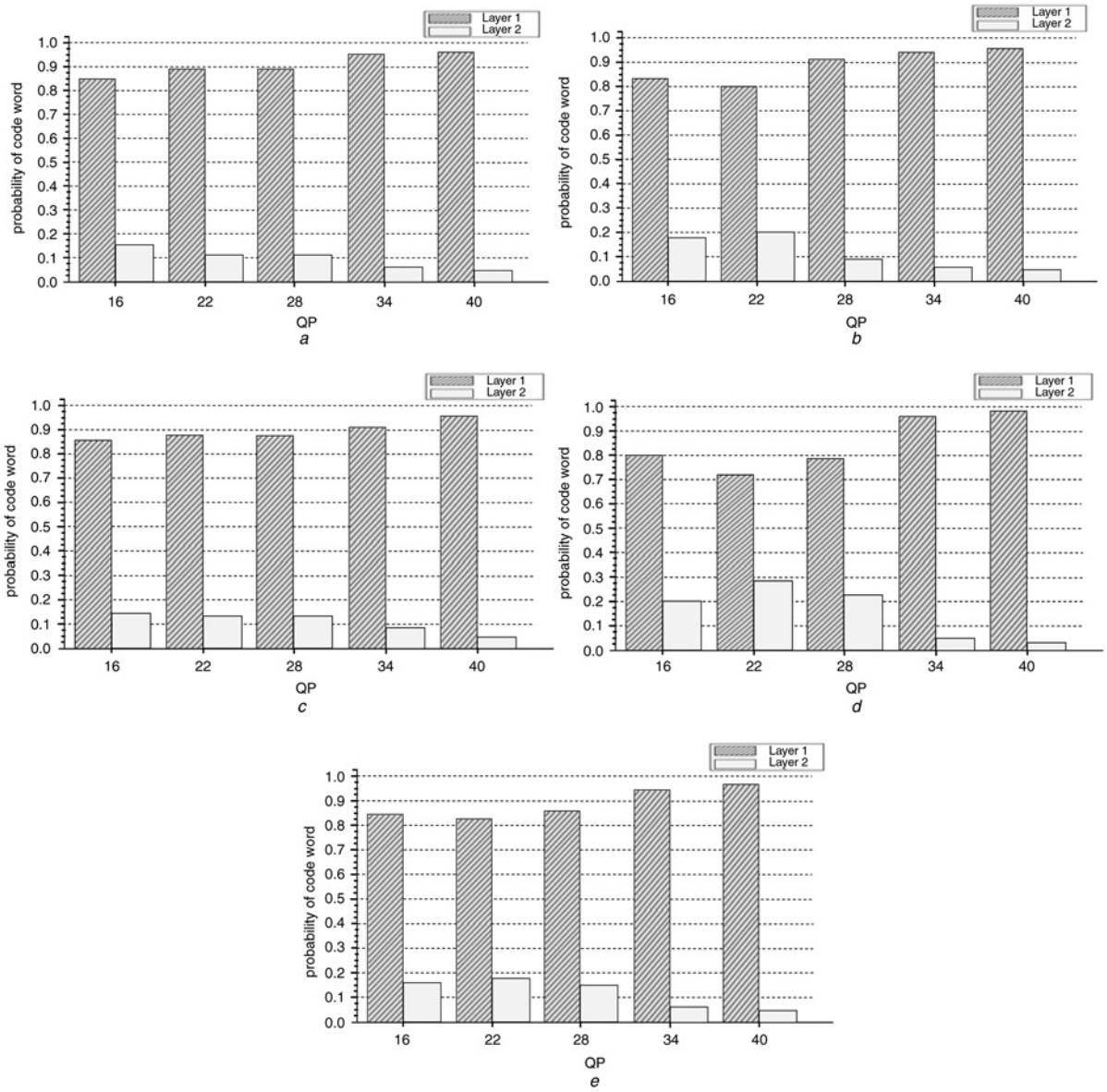

Fig. 3 *Probabilities of layer with the proposed two-layer decoding with IPPPP frames*

a CIF 30 frames/s 'Foreman' sequence
b CIF 30 frames/s 'Mobile' sequence
c CIF 30 frames/s 'Akiyo' sequence
d CIF 30 frames/s 'Coastguard' sequence
e Average probability of the four CIF 30 frames/s sequences

the codewords whose code length is less than 8 to the first layer of the CT module by hit statistics [8, 9]. Table 1 shows that the proposed architecture can reduce the codeword numbers in the first layer of the CT module by 74% compared with Lin's design [8, 9]. This design attains the low-power goal because the number of codewords with high decoding probability is small, and these codewords are concentrated on the first layer decoding to reduce the probability of requiring the second layer decoding.

The parallel decoding in the second layer prevents the accumulation of E_{M_i} and minimises the power consumption of searching misses. Even though a decoding process of the first layer fails to search a pattern, it just loses a low E_{M_1} and the overload energy. If a video sequence is encoded with a low QP value, its video quality is good enough and the decoding probability of the second layer increases accordingly. Owing to the parallel decoding of the second layer, the power consumption can be converged by the minimum power consumption of searching misses. This stabilises the power consumption of the system even when the QP varies. The optimal power model of the proposed two-layer architecture can be described as follows

$$\begin{aligned} E_{\text{optimal}} &= P_{L_1}E_{L_1} + P_{L_2}(E_{L_2} + E_{\text{prefix}} + E_{M_1}) \\ &= P_{L_1}\left(\sum_{i=1}^{n_1} P_{L_1T_i}\sum_{j=1}^{m_{1,i}}(P_{C_j}E_{L_1T_iC_j}) + E_{\text{MUX}}\right) \\ &\quad + P_{L_2}\left(\sum_{i=1}^{n_2} P_{L_2T_i}\sum_{j=1}^{m_{2,i}}(P_{C_j}E_{L_2T_iC_j})\right. \\ &\quad \left. + E_{\text{prefix}} + E_{\text{MUX}} + E_{M_1}\right) \end{aligned} \tag{4}$$

where P_{L_1} and P_{L_2} are the decoding probability of the first and

© The Institution of Engineering and Technology 2012

doi: 10.1049/iet-ipr.2009.0344

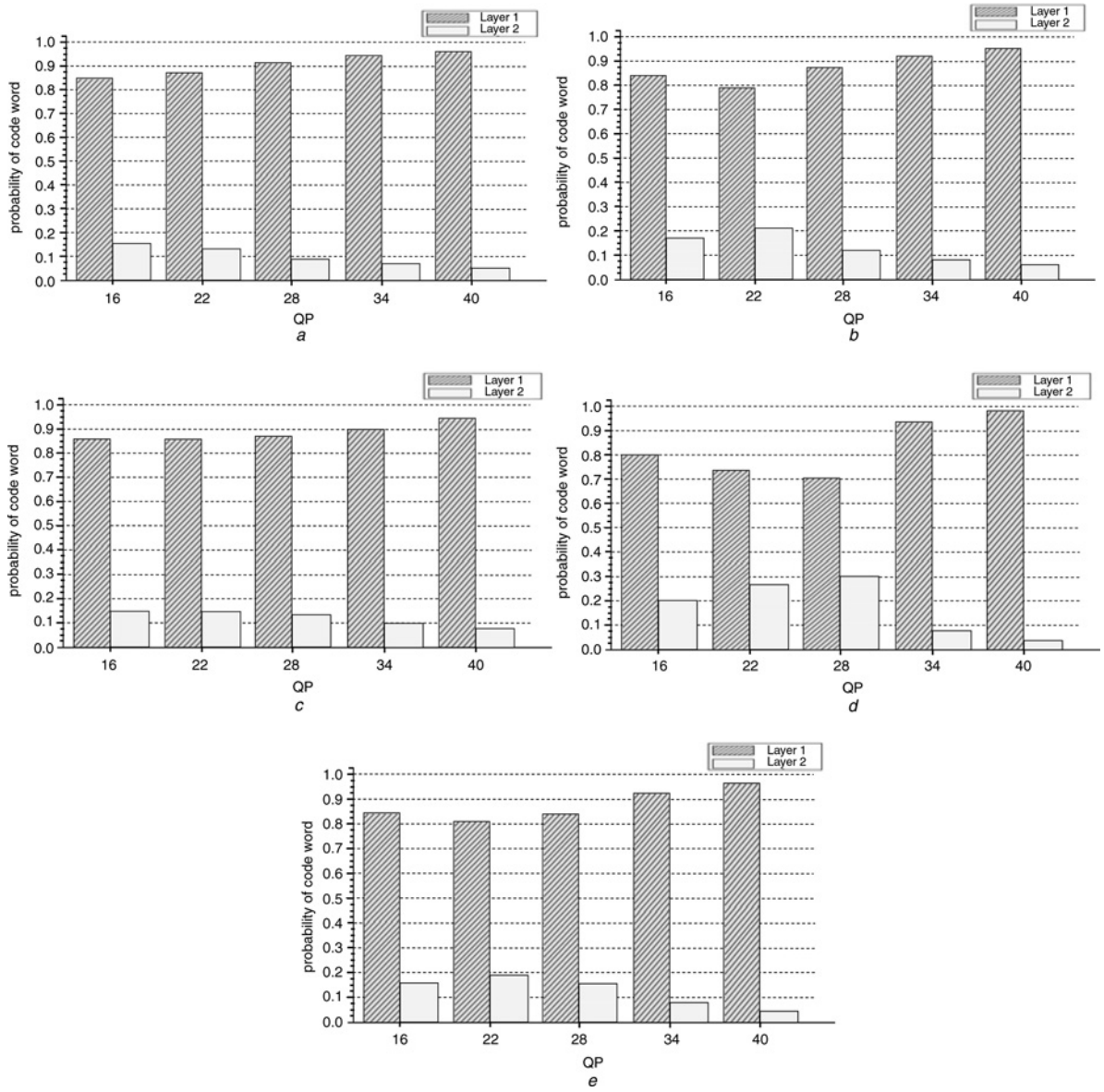

Fig. 4 *Probabilities of layer with the proposed two-layer decoding with IBPBP frames*

a CIF 30 frames/s 'Foreman' sequence
b CIF 30 frames/s 'Mobile' sequence
c CIF 30 frames/s 'Akiyo' sequence
d CIF 30 frames/s 'Coastguard' sequence
e Average probability of the four CIF 30 frames/s sequences

second layers, respectively, E_{L_1} and E_{L_2} are the energy consumption of the first and second layers, respectively, n_1 and n_2 mean the numbers of the LUTs in the first and second layers, respectively, $m_{1,i}$ and $m_{2,i}$ mean the numbers of codewords in the ith LUT in the first and second layers, respectively, P_{C_j} denotes the searching probability of the jth codeword in the LUT, $P_{L_N T_i}$ is the probability of the ith LUT which is used in the first or the second layer for $N = 1$ or 2, $E_{L_N T_i C_j}$ is the searching energy of the jth codeword in the ith LUT for decoding in the first or the second layer for $N = 1$ or 2, and E_{MUX} and E_{prefix} are the multiplexer and overload energy consumption for the decoding.

3.2 Look-up table mergence

Section 3.1 uses the proposed TLS algorithm to reduce the power consumption of the CAVLD for the low-power design. However, the area cost of the direct LUT design is still too large in the second layer decoding. To solve this issue, this study uses the LTM technique in [13, 15, 16] to reduce the hardware cost of the LUTs in the second layer decoding.

Owing to the variable length decoding in the CAVLD, no fixed boundary exists between different codewords. This leads to the issue that the related code length must be decoded correctly; otherwise, the continued decoding process will fail. Thus, when we search the LUTs for decoding, an important decoding output is the code length of the codeword. When a code length is decoded, different LUTs may have the same decoding result. For instance, the symbol decoding of the codeword '00010' in the LUT0 and the LUT1 may be different, but the code length decoding of this codeword in the LUT0 and the LUT1 can obtain the same value, that is, 5. The proposed design uses the LTM technique to combine the parts of the LUT, which can be

© The Institution of Engineering and Technology 2012

Table 1 Codeword numbers of the first layer in CT

	LUT0 ($0 \leq nC < 2$)	LUT1 ($2 \leq nC < 4$)	LUT2 ($4 \leq nC < 8$)
Lin [8] and Lin *et al.* [9]	13	23	40
proposed	5	7	8

decoded to become the same code length and symbols. The LTM technique can be applied to the H.264/AVC CAVLD design and other low-cost variable LUT designs.

Suppose that a module includes three different LUTs, and their original contents are shown in Table 2. From the observation of the three LUTs, although the first six symbols are different, their corresponding codewords are the same. Thus, it is possible to merge the first six codewords among the three LUTs. Owing to the table mergence, the codeword number is reduced from 24 to 12. This results in a low-cost LUT design. Table 3 shows the optimised results of Table 2 with the LTM technique. The proposed CAVLD design uses the LTM method to reduce the LUT area cost of the CT, TZ and RB modules, which have higher hardware costs than the other modules. In Table 4, the look-up table-based parts (i.e. CT, TZ and RB) in the symbol decoder of the CAVLD show that the proposed LTM design requires smaller gate counts than the other schemes in [8, 18] for low-power and low-cost goals. Compared with the convenient design, the proposed LTM design can reduce the area cost by 81.1%. Meanwhile, the area reduction efficiency in our design is similar to that in [15], which is about 88.1%.

4 Functional verification, VLSI implementation and comparison

4.1 Functional verification

Fig. 5 depicts the overall architecture of the proposed CAVLD design. The CT module is separated into two-layer decoding. To verify the functional correctness of the proposed CAVLD circuits, the hardware/software co-simulation method is used

Table 2 Original LUT before the LTM method is used

LUT0			LUT1			LUT2		
Codeword	Symbol	Length	Codeword	Symbol	Length	Codeword	Symbol	Length
111	A	3	111	I	3	111	Q	3
110	B	3	110	J	3	110	R	3
0101	C	4	0101	K	4	0101	S	4
0100	D	4	0100	L	4	0100	T	4
0001 1	E	5	0001 1	M	5	0001 1	U	5
0001 0	F	5	0001 0	N	5	0001 0	V	5
0000 11	G	6	0000 1	O	5	0000 0	W	5
0000 01	H	6	0000 00	P	6	0000 1	X	5

Table 3 Optimised LUT of Table 2 with the LTM technique

Common		LUT0		LUT1		LUT2	
Codeword	Length	Codeword	Length	Codeword	Length	Codeword	Length
111	3	0000 11	6	0000 1	5	0000 0	5
110	3	0000 01	6	0000 00	6	0000 1	5
0101	4						
0100	4						
0001 1	5						
0001 0	5						

Table 4 Hardware cost profile of symbol decoder in different CAVLD (gate count)

Item			Methods		
			Design in [8, 9]	Design in [18]	Proposed
symbol decoder	LUT-based design	CT	1037	1825	723
		TZ	843		470
		RB	226		155
	arithmetic-based design	Level	662	665	1834
		Trailing_ones	103	102	

© The Institution of Engineering and Technology 2012

doi: 10.1049/iet-ipr.2009.0344

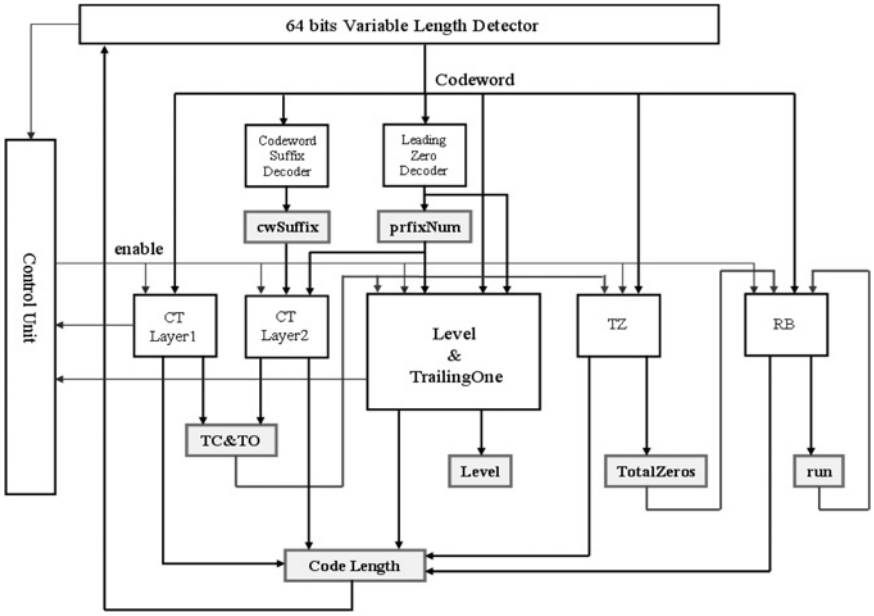

Fig. 5 *Overall architecture of the proposed CAVLD*

Table 5 Gate count result of the proposed CAVLD with TSMC 0.18 μm process

Conditions	Module							
	CAVLD							
	LUT table			Level and tone	Prefix/suffix decoder	Control unit	64-bit buffer detector	Total
	CT	TZ	RB					
at 100 MHz	715	454	142	1454	367	763	1471	5366
at 125 MHz	723	470	155	1834	387	843	2021	6433

to emulate its function. At the software part in the co-simulation, the H.264 reference software version JM 10.2 [22] is used for the system simulation and integration of the H.264/AVC codec. At the hardware part in the co-simulation, the SMIMS VeriEnterprise® FPGA platform [23] is applied to emulate the function of the proposed CAVLD with 48 MHz operational frequency, where the applied FPGA device is a Xilinx Virtex-4 XC4LVX60 chip.

In the co-simulation environment, we only disable the internal functions of the subroutine of the CAVLD part in the JM 10.2 software and replace them with the hardware design in the FPGA emulation platform for the CAVLD decoding. When the JM program in the personal computer calls the CAVLD subroutine, the un-decoded bit stream at the input port of the subroutine is bypassed into the FPGA chip via the USB and the related data of residual blocks are decoded in the chip. After the residual blocks are processed and decoded by hardware, the decoded data are stored in the first in, first out (FIFO) and the interrupt signal is sent to the software side via the USB for returning the decoded data to the personal computer. The JM program then continues its decoding processes to obtain the reconstructed video. Thus, the functional verification can be completed for our CAVLD design.

Table 6 Total gate count of the whole CAVLD system with TSMC 0.18 μm process

Conditions	Module		
	CAVLD	Output registers	Total
at 100 MHz	5366	3095	8461
at 125 MHz	6433	3100	9533

4.2 VLSI implementation and comparison

The proposed low-power and cost-effective CAVLD architecture was implemented using the cell-based design flow with 0.18-μm TSMC CMOS technology. For the purpose of the functional design, simulation and verifying the correction of the hardware implementation, the proposed architecture was written with Verilog HDL codes, and the ModelSim™ EDA tool was used for the functional simulation. Then the Synopsys Design Compiler™ was used to synthesise the Verilog HDL codes. Therefore the HW/SW co-simulation, which is composed of H.264 reference software version JM10.2 and Xilinx Virtex-4 FPGA platform, was also applied to verify the functional correctness. Table 5 shows the gate count results, where our architecture is synthesised at 100 and 125 MHz, respectively. Table 6 shows the total gate count of the whole proposed CAVLD system, which includes the CAVLD core and output registers. Table 7 shows the VLSI performance comparison with the previous high-throughput CAVLDs. Table 7 shows that the total gate count of the

© The Institution of Engineering and Technology 2012

Table 7 VLSI performance comparison with different high-throughput CAVLDs

Specifications	Lin *et al.* [18]	Yu and Chang [14]	Tsai and Fang [19]	Yeo and Shin [20]	Lee *et al.* [21]	Proposed
technology, μm	0.18	0.18	0.18	0.25	0.18	0.18
maximum frequency	213 MHz	125 MHz	160 MHz	N/A	125 MHz	125 MHz
area: CAVLD logic (gate count)	4308	13 192	9312	N/A	N/A	6433
area: output buffer (gate count)	2503	N/A	3877	N/A	N/A	3100
total area (gate count)	6883	13 192	13 189[b]	12 484[a]	14 723	9553
throughput (symbol/clock)	>1	>1	>1	>1	>1	>1

[a]Assume that one gate requires 10 μm^2 chip area
[b]Total gate count is estimated at 125 MHz operational frequency

Table 8 VLSI performance comparison with different low-power low-cost CAVLDs

Specifications	Lin [8] and Lin *et al.* [9]	Lin *et al.* [18]	Proposed
technology, μm	0.18	0.18	0.18
maximum frequency, MHz	125	213	125
area: CAVLD logic (gate count)	4758	4308	6433
area: output buffer (gate count)	3718	2503	3100
total area (gate count)	8476	6883	9553
throughput (symbol/clock)	1	>1	>1

proposed CAVLD was smaller than that of the works in [14, 19, 20, 21], but larger than that in [18]. The high-throughput design can provide a larger symbol per clock than 1, where the decoding symbol means the syntax element, that is Coeff_Token, Trailing_One, Level, TZ or RB. Table 8 shows the VLSI performance comparison with the previous low-power low-cost works. In Table 8, the total gate count of the proposed CAVLD is larger than that of the works in [8, 9, 18], but the throughput in our design is larger than that in [8, 9]. Although the hardware cost in our design is a little larger than that in [8, 9, 18], the proposed CAVLD is cost-effective to achieve the purpose of the low power consumption. Table 9 shows the VLSI performance comparison with the previous cost-effective designs. From Table 9, our cost-effective design can not only achieve low-power consumption but also provide a larger throughput than the previous cost-effective designs in [6, 7].

Table 10 shows the decoding performance (i.e. cycles/MB) of the proposed decoder with 30 frames/s sequences at QP = 28. In Table 10, the cycles per macroblock for our CAVLD decoding are evaluated by ModelSim™ HDL simulator for cycle-based functional simulation and the input test vectors are generated from JM10.2 software.

Table 9 VLSI performance comparison with different cost-effective CAVLDs

Specifications	Wu *et al.* [6]	Chang *et al.* [7]	Proposed
technology, μm	0.25	0.18	0.18
maximum frequency, MHz	125	125	125
area: CAVLD logic (gate count)	6100	4720	6433
area: output buffer (gate count)	N/A	2793 (predict register 2430)	3100
total area (gate count)	N/A	9943	9553
throughput (symbol/clock)	1	1	> 1

Table 10 Comparison of the cycles per macroblock for CAVLC decoding with 30 frames/s sequences at QP = 28

	SVA_BA1_B	BA1_Sony_D	Foreman	Mobile
design [7] in [14][a]	N/A	N/A	192	395
Yu and Chang [14][a]	N/A	N/A	53	174
proposed[b]	121	163	120	347

[a]Test video sequence is the QCIF size
[b]Test video sequence is the CIF size

Only I-frames in all the sequences were used in the simulations, because decoding I-frames needs more working cycles per MB than decoding B-frames or P-frames in the CAVLD processing. In most of the test sequences, when QP is set to 28, the reconstructed video quality and bit rate are acceptable for applications, and then the analysis of decoding processing under this environment is reasonable. The proposed design requires smaller decoding cycles per MB than Chang's design, but Yu's design in [14] needs the smallest processing cycles among the three CAVLD architectures. If the operation frequency is set to 6.2 MHz, our design can decode a CIF format 4:2:0 30 frames/s video sequence. If the operation frequency is set to 25 MHz, our design can decode a 4CIF format 4:2:0 30 frames/s video sequence. Thus, the result shows that the proposed architecture can meet the demand of the real-time video decoding for H.264/AVC portable applications.

4.3 Power estimation and comparison

This study measured power consumption under the same operating conditions as in [8, 9]. These operating conditions are described as follows. The temperature was 25°C, the voltage was set to 1.8 V and the operation frequency was 50 MHz. The PrimePower™ EDA tool was used to estimate the power consumptions. Under the baseline intra mode and four different QP values, that is 16, 22, 28 and 34, the power consumption of the CAVLD was measured with four CIF format 30 frames/s video sequences. Fig. 6 shows the power comparison with Lin's design [8, 9]. The 'original' design uses the direct and non-optimal architecture for implementation. Fig. 6 shows that the original and Lin's architectures have the largest variance in power consumption among the different QPs. However, the proposed architecture maintains stable power consumption on different QPs. Table 11 shows the average power consumptions of the proposed design and Lin's design [8, 9]. The average power consumption of the proposed design is 44–48% better than that of Lin's design under the same CIF 30 frames/s I-frame sequences [8, 9]. A comparison of

© The Institution of Engineering and Technology 2012

doi: 10.1049/iet-ipr.2009.0344

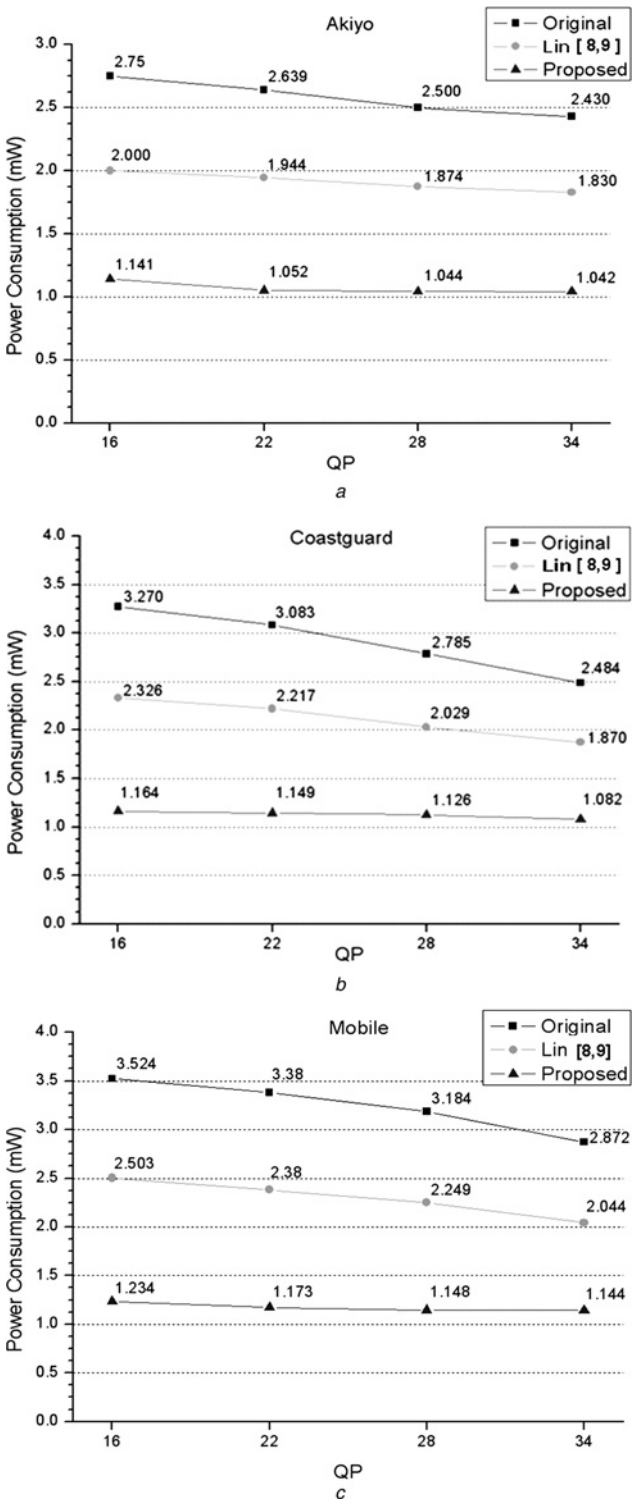

Fig. 6 *Comparison of power consumption among different architectures*

a Power consumption of video sequence 'Akiyo' with 30 frames/s CIF format

b Power consumption of video sequence 'Coastguard' with 30 frames/s CIF format

c Power consumption of video sequence 'Mobile' with 30 frames/s CIF format

the proposed design with the conventional CAVLD design shows that the proposed design reduces the average power consumption by 58%. However, the designs in [8, 9, 18] reduce the average power consumption by 25 and 40%, respectively.

Table 11 Comparison of the average power consumption with CIF 30 frames/s I-frame sequences under QP = 16, 22, 28, and 34 with 0.18 μm process

Average power consumption, mW			
Architecture	Sequence		
	Akiyo	Coastguard	Mobile
Lin [8] and Lin *et al.* [9]	1.912	2.111	2.294
proposed	1.069	1.130	1.179
power reduction, %	44.05	46.46	48.60

5 Conclusion

This study proposes an efficient low-power and cost-effective CAVLCD for H.264/AVC portable applications. To obtain the low-power goal, this study uses the two-phase decoding architecture of LUT instead of the traditional hierarchy scheme. The first layer decoding achieves the low decoding energy by introducing the codeword properties, and it reduces the most power consumption according to the high decoding probability. The second layer decoding is implemented by the parallel decoding, which avoids the accumulation of the miss energy when the first layer fails to decode the codeword. To achieve a cost-effective design, the common code words are merged to reduce the hardware cost of different LUTs in the second layer decoding. The proposed design is based on TSMC 0.18-μm CMOS technology. This study also successfully verifies the proposed CAVLD on the Xilinx FPGA emulation platform. The simulation results of the power consumption show that the total average power consumption of the proposed design can be reduced from 44 to 48% more than the previous low-power CAVLCD proposed by Lin [8] and Lin *et al.* [9]. A comparison of the proposed design with the conventional CAVLD design shows that the proposed design reduces the average power consumption by 58%.

6 Acknowledgments

This work was supported in part by the National Science Council, Taiwan, R.O.C., under Grant NSC99-2221-E-005-115. The authors would like to thank the anonymous reviewers whose careful reviews and detailed comments helped improve the readability of this paper, and the National Chip Implementation Center (CIC) in Taiwan for providing the synthesis environment and related EDA tools.

7 References

1 Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG: 'Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC)', 2003

2 Lei, S.M., Sun, M.T.: 'An entropy coding system for digital HDTV applications', *IEEE Trans. Circuits Syst. Video Technol.*, 1991, **1**, (1), pp. 147–151

3 Molloy, S., Jain, R.: 'Low power VLSI architecture for variable-length encoding and decoding'. 40th Midwest Symp. on Circuits and Systems, August 1997, vol. 2, pp. 997–1000

4 Cho, S.H., Xanthopoulos, T., Chandrakasan, A.P.: 'A low power variable length decoder for MPEG-2 based on nonuniform fine-grain table partitioning', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 1999, **7**, (2), pp. 249–257

5 Lee, S.W., Park, I.C.: 'A low-power variable length decoder for MPEG-2 based on successive decoding of short codewords', *IEEE*

© The Institution of Engineering and Technology 2012

Trans. Circuits Syst. II: Analog Dig. Signal Process., 2003, **50**, (2), pp. 73–82
6 Wu, D., Gao, W., Hu, M.Z., Ji, Z.Z.: 'A VLSI architecture design of CAVLC decoder'. Fifth Int. Conf. on ASIC, October 2003, vol. 2, pp. 962–965
7 Chang, H.C., Lin, C.C., Guo, J.I.: 'A novel low-cost high performance VLSI architecture for MPEG-4 AVC/H.264 CAVLC decoding'. IEEE Int. Symp. on Circuits and Systems, 2005, vol. 6, pp. 6110–6113
8 Lin, H.Y.: 'Low power context-based adaptive variable length decoder for H.264 video decoders'. Master thesis, National Cheng Kung University, Taiwan, 2006, p. 76
9 Lin, H.Y., Lu, Y.H., Liu, B.D., Yang, J.F.: 'Low power design of H.264 CAVLC decoder'. IEEE Int. Symp. on Circuits and Systems, May 2006, pp. 2689–2692
10 Kim, Y.H., Yoo, Y.J., Shin, J., Choi, B., Paik, J.: 'Memory efficient H.264/AVC CAVLC for fast decoding', *IEEE Trans. Consum. Electron.*, 2006, **52**, (3), pp. 943–952
11 Moon, Y.H., Kim, G.Y., Kim, J.H.: 'An efficient decoding of CAVLC in H.264/AVC video coding standard', *IEEE Trans. Consum. Electron.*, 2005, **51**, (3), pp. 933–938
12 Wen, Y.N., Wu, G.L., Chen, S.J., Hu, Y.H.: 'Multiple-symbol parallel CAVLC decoder for H.264/AVC'. IEEE Asia Pacific Conf. on Circuits and Systems, December 2006, pp. 1240–1243
13 Liu, T.M.: 'Study of MPEG-2 and H.264 video decoders for mobile applications'. PhD dissertation, National Chiao Tung University, May 2007
14 Yu, G.S., Chang, T.S.: 'A zero-skipping multi-symbol CAVLC decoder for MPEG-4 AVC/H.264'. IEEE Int. Symp. on Circuits and Systems, 2006, pp. 5583–5586
15 Sun, S.M., Liu, T.M., Lee, C.Y.: 'A self-grouping and table-merging algorithm for VLC-based video decoding system'. IEEE Asia Pacific Conf. on Circuits and Systems, December 2006, pp. 1567–1570
16 Lee, W.C., Li, Y., Lee, C.Y.: 'Design of a memory-based VLC decoder for portable video applications'. IEEE Asia Pacific Conf. on Circuits and Systems, December 2008, pp. 1340–1343
17 Huang, H.J., Fan, C.P.: 'Architecture design of low-power and low-cost CAVLC decoder for H.264/AVC'. IEEE Asia Pacific Conf. on Circuits and Systems, December 2008, pp. 1336–1339
18 Lin, H.Y., Lu, Y.H., Liu, B.D., Yang, J.F.: 'A highly efficient VLSI architecture for H.264/AVC CAVLC decoder', *IEEE Trans. Multimed.*, 2008, **10**, (1), pp. 31–42
19 Tsai, T.H., Fang, D.L.: 'An efficient CAVLD algorithm for H.264 decoder'. Int. Conf. on Consumer Electronics, January 2008, pp. 1–2
20 Yeo, D., Shin, H.: 'Enhanced parallel decoding for H.264/AVC CAVLC by using precomputation'. Int. SoC Design Conf., November 2008, vol. 2, pp. 93–96
21 Lee, G.G., Lo, C.C., Chen, Y.C., Lei, S.F., Lin, H.Y., Wang, M.J.: 'Low complexity and high throughput VLSI architecture for AVC/H.264 CAVLC decoding'. IEEE Int. Symp. on Circuits and Systems, May 2009, pp. 1229–1232
22 Joint Video Team (JVT) reference software JM10.2
23 SMIMS Technology Corp, http://smims.com/en/

© The Institution of Engineering and Technology 2012

doi: 10.1049/iet-ipr.2009.0344